PATENT
ATTY. DOCKET NO. IBM/67DV1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Ex parte Galbraith et al.

Appeal No. _____

Serial No.:	**09/935,939**
Filed:	**August 23, 2001**
Group Art Unit:	**2187**
Examiner:	**Brian Peugh**
Applicant:	**Galbraith et al.**
Title:	**ADVANCED READ CACHE MANAGEMENT**

Cincinnati, Ohio 45202

April 24, 2003

BRIEF ON APPEAL

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, postage prepaid in an envelope addressed to: Assistant Commissioner of Patents, Box AF FEE, Washington, D.C. 20231 on: April 24, 2003

Laura G. Koehler

Laura G. Koehler

This brief is in furtherance of Applicant's Notice of Appeal filed February 24, 2003, appealing the decision of the Examiner dated October 23, 2003 finally rejecting claims 21-30 and 45-57. A copy of the claims appears in the Appendix to this brief. This brief is transmitted in triplicate.

Real Party In Interest

The real party in interest in this appeal is INTERNATIONAL BUSINESS MACHINES CORPORATION, a corporation of New York having a place of business at New Orchard Road, Armonk, New York 10504.

Related Appeals and Interferences

There are no such appeals or interferences.

Status of Claims

Claims 21-24, 30, 45-48 and 54 stand rejected under 35 U.S.C. § 103(a), asserted to be unpatentable over the combination of McNutt et al. (U.S. Patent 5,649,153) in combination with Johnson et al. (U.S. Patent 5,577,236). Claims 25 and 49 stand rejected under 35 U.S.C. § 103(a), asserted to be unpatentable over the combination of McNutt et al. in combination with Johnson et al. and further in combination with Mayfield (U.S. Patent 5,737,565). Claims 55-57 stand rejected under 35 U.S.C. § 103(a), asserted to be unpatentable over the combination of McNutt et al. in combination with Johnson et al. and further in combination with the book by Tanenbaum, Structured Computer Organization.

Claims 26-29 and 50-53 are objected to as being dependent upon a rejected base claim, but are otherwise deemed allowable.

Claims 21-30 were originally filed with the application. Applicant's Amendment of September 12, 2002 added claims 45-57, and amended claims 21-23. An Amendment for Appeal submitted with this brief proposes correction of a typographical error in claims 51-52.

Status of Amendments

The Amendment for Appeal is pending as of the date of filing of this brief. The claims in the appendix reflect the assumed entry of this amendment.

Summary of Invention

This invention relates to the practice of caching data in computer systems. In computer systems, data is frequently stored in hard disk drives or other direct access storage devices (DASD's) that are relatively high capacity but relatively slow speed. To improve access speed, a cache is often used in connection with a DASD. The cache includes relatively high speed memory that stores copies of data from a DASD, so that data can be accessed more rapidly than would be required using the DASD alone.

The invention relates to a novel cache management method, and a circuit and program product that performs that method. The cache management method permits run-time reconfiguration of the size of a cache memory, so that cache memory may be added and removed, without requiring computer system downtime.

Previously, reconfiguration of cache memory required downtime; the cache needed to be purged entirely of data, and then the entire computer system needed to be deactivated, only after which the physical memory used by the cache could be changed (increased or decreased). The features of the present invention avoid the need for downtime for such an operation, and thus avoid a substantial opportunity cost caused by computer system downtime, that was heretofore inherent in upgrading or changing the cache hardware of a computer system.

Details of the methods performed in accordance with the present invention are provided in the specification beginning at page 48, with reference to Figs. 7 and 8.

The claims presented in this application recite this novel aspect. Specifically, each of those claims recite a "cache memory" having a "total memory capacity", which is storing "copies of data retrieved from [a] direct access storage device" as well as "a cache directory identifying data in [the] direct access storage device". The claims thus recite a situation in which there is a existing cache having a defined "total memory capacity", and which is storing data. The claims further recite normal use of this cache, specifically, responding to "a request for access" by accessing a copy in the cache or accessing a storage device, based upon whether desired data is available in the cache memory. Finally, the claims recite the step of "responding to a change in said total available cache memory capacity by altering said cache directory". This last step sets forth the feature of an affirmative ability to respond to changes in "total available cache memory", by the affirmative act of "altering said cache directory".

Issues

Whether the subject matter of any of claims 21-30 and 45-57 is obvious in light of McNutt et al., Johnson et al., Mayfield or Tanenbaum.

Grouping of Claims

The claims do not stand or fall together. Claims 26-29 and 50-53 are deemed allowable by the Examiner, and therefore do not stand or fall with the rejected independent claims.

Argument

There does not appear to be dispute as to the novelty of the concept of a cache control method, circuit or program that will respond to run-time changes in cache memory capacity. Indeed, the Examiner appears to concede that such is novel, but disputes the interpretation of the claim language as reciting this concept.

In the following discussion, Applicant will first demonstrate why the prior art cited by the Examiner fails to disclose responding to run-time changes to cache memory capacity. Then, Applicant will discuss the proper interpretation of the claim language.

The Examiner's base rejection is that McNutt et al., combined with Johnson, show all elements of the independent claims. Applicant will explore the content of these references.

McNutt et al. describes an algorithm for managing a cache, specifically, an algorithm that caches data on either a "record" basis or a "track" basis, as appropriate.

While McNutt et al. thus deals with management of a cache, nowhere does McNutt et al. mention the possibility of changes in total memory capacity of a cache.

The Examiner has noted that, as data is stored in or deleted from the McNutt et al. cache, the amount of "free" space in that cache will change. However, the claims recite a change in "total available cache memory capacity", not changes in "free" space. Nothing in McNutt et al. anticipates a change in "total available cache memory capacity". The Examiner has conceded this point, stating (page 3 of the Final Action) "the difference between claimed subject matter and that of McNutt et al. is that the claims ... are directed toward a change in the overall total memory of a cache...".

The Examiner uses Johnson to fill the gaps in McNutt et al. Johnson shows a memory controller that, as is conventional, utilizes modular memory. Specifically, Johnson's memory is contained in Single Inline Memory Modules (SIMM's). The Examiner's comment is that Johnson's SIMM's can be added and removed, and the memory controller must account for the amount of total memory resulting from the number of SIMM's that are present.

Applicant notes, first, that SIMM's can be added or removed from Johnson's system only by powering down, adding or removing SIMM's, powering up, and then reconfiguring. (This is, indeed, typical of all computer systems having reconfigurable memory.) Johnson states at col. 8, lines 33-45 that "flash memory (not shown) is used to store detailed information concerning the memory components utilized by the system. .... Such information, for example, may include the number of SIMMs present, the memory capacity of each SIMM [etc.] ... This data is usually written by a user to flash memory

when initializing the computing system that hosts the memory. The flash memory typically comprises a write-once-read-many ("WORM") memory component such as an electrically programmable read only memory ("EPROM")." [emphasis added]

Johnson thus only suggests that physical memory can be configured "when initializing the computing system that hosts the memory". Johnson does not suggest, and indeed teaches away, from run-time reconfiguration of total available memory capacity, e.g., due to a change in "total available memory capacity". The only way Johnson's system can change total available memory capacity, is to restart the entire computer system, writing new information into flash memory "when initializing the computing system that hosts the memory."

Building on this observation, it should be noted that, as part of powering down a computer system, all information stored in a cache must first be purged (and written back, if needed) and the cache essentially deactivated; otherwise, there would be data loss. Thus, the reconfiguration process that would be followed in Johnson, or any McNutt et al./Johnson combination, would be to

1. purge the cache of all information,
2. deactivate the cache, and the computer system itself,
3. reconfigure physical memory used by the cache,
4. restart the computer system, creating a new memory configuration "when initializing the computing system",
5. reactivate the cache.

This compares to the claimed process, in which a "change ... in total available cache memory capacity" triggers a response that simply "alter[s the] cache directory". A McNutt et al./Johnson combination cache system would not "respond[] to a change in memory" or "alter[a] cache directory" in response to such a change. Rather, the user

would be required to completely deactivate both the cache system and the computer, before any memory change occurs. There is no function to "respond[] to a change in memory"; if the user were to pull out a SIMM from the Johnson computer system while the computer were still running, an error would be a certain result.

The Examiner appears to dispute whether the claims bring forth these distinctions from McNutt et al. and Johnson. Specifically, the Examiner's Advisory Action states that "Applicant's arguments are directed toward real-time memory capacity updating, although the dynamic or real-time feature is not claimed". Applicant disagrees.

As noted above, each of the claims recite a "cache memory" having a "total memory capacity", which is storing "copies of data retrieved from [a] direct access storage device" as well as "a cache directory identifying data in [the] direct access storage device". The claims thus recite a precedent situation in which there is a defined cache having a defined "total memory capacity" and which is storing data. The claims further recite normal use of this defined cache, specifically, responding to "a request for access" by accessing a copy in the cache or accessing a storage device, based upon whether a copy of the data to be accessed is available in the cache memory. Finally, the claims recite the step of "responding to a change in said total available cache memory capacity by altering said cache directory". This last step thus sets forth that management of the cache includes an affirmative response to changes in "total available cache memory" and the affirmative act of "altering said cache directory" as a consequence.

Applicant submits that, in a McNutt et al./Johnson combination, as discussed above, "total available memory capacity" changes would only proceed by deactivating the

cache system, as well as the entire computer. In such a scenario, there is no step of "altering [a] cache directory" – the "cache directory" that existed, ceases to exist as part of the deactivation of the cache and computer. When the computer is restarted, the cache directory must be rebuilt entirely as the cache system is reactivated. It is not "alter[ed]" from a preceding condition to a new condition that reflects the change in memory; it ceases to exist and then is recreated in a different form. In contrast, the claims recite a preceding condition in which the "cache directory" is stored in a cache memory having a "total available memory capacity", there is a "change in said total available memory capacity", in response to which there is "altering" of the pre-existing cache directory.

With regard to nonobviousness, Applicant submits that the present claims provide a far more fluid ability to adjust to changes in memory configuration than was possible in the prior art. Specifically, Applicant's cache management approach permits run-time reconfiguration of the cache size, so that cache memory may be added and removed in real time without requiring computer system downtime. This feature avoids the opportunity cost that was previously inherent in upgrading or changing the cache hardware of a computer system, as the computer system need not be brought off-line (and rendered temporarily unproductive) to change total available cache memory capacity.

Applicant respectfully submits that the Examiner's Final Action, and advisory action, improperly rejects and misreads the claims. Neither the McNutt et al. nor the Johnson patents disclose what the claims recite, a system that can "respond[] to a change in ... total available memory capacity". Rather, the combination of the McNutt et al. and

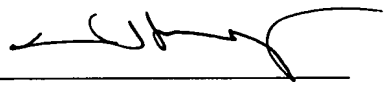
Johnson references requires (as is typical) that power be removed to effect a physical memory change.

The Examiner's citation to Mayfield is strictly for showing the use of LRU queuing in a cache. Nothing in Mayfield fills the gaps in McNutt et al. and Johnson noted above.

The Examiner's citation to Tanenbaum is strictly for the assertion that hardware and software are logically equivalent, and the manner in which software may be stored. Nothing in Tanenbaum fills the gaps in McNutt et al. and Johnson noted above.

In view of the foregoing, and in view of the clear patentable distinction of all claims over the cited references, Applicant submits that all claims are allowable. Accordingly, Applicant submits that the Examiner's rejection is in error and a reversal of the rejection and allowance of the claims is therefore requested.

Respectfully submitted,
Wood, Herron & Evans, L.L.P.

By 
Thomas W. Humphrey
Reg. No. 34,353

2700 Carew Tower
441 Vine Street
Cincinnati, OH 45202-2917

Voice: (513) 241-2324
Facsimile: (513) 241-6234

APPENDIX

21. (Once Amended) A method of caching data for a direct access storage device having a plurality of addressable locations, comprising the steps of:

storing, in a cache memory having a total available memory capacity, copies of data retrieved from said direct access storage device, and a cache directory identifying data in said direct access storage device for which copies are stored in said cache memory, and identifying memory locations in said cache memory where each said copy is stored,

responding to a request for access to a storage device location for which a copy is stored in the cache memory, by accessing the copy stored in the cache memory,

responding to a request for access to a desired storage device location for which a copy is not stored in the cache memory, by accessing said desired storage device location from said storage device, and

responding to a change in said total available cache memory capacity by altering said cache directory.

22. (Once Amended) The method of claim 21 wherein in response to an increase in the total cache memory capacity, the method further comprises modifying said cache directory to identify memory locations in said cache memory where copies of data from said direct access storage device may be stored, and then storing, in said cache memory, copies of data retrieved from said direct access storage device.

23. (Once Amended) The method of claim 21 wherein in response to a reduction in the total cache memory capacity, the method further comprises modifying said cache directory to no longer identify memory locations that are not available in said cache memory.

24. The method of claim 21 further comprising monitoring accesses to data for which copies are stored in the cache memory.

25. The method of claim 24 wherein monitoring accesses to data comprises maintaining a LRU queue in which data are ordered from most to least recently used.

26. The method of claim 25 wherein monitoring accesses to data comprises maintaining statistics on types of accesses made to data.

27. The method of claim 26 wherein maintaining statistics on types of accesses made to data comprises maintaining a counter associated with blocks of data, said counters being credited or penalized in response to types of accesses made to the associated block of data.

28. The method of claim 27 wherein maintaining statistics further comprises crediting a counter by a predetermined credit in response to a read to a block of data associated with said counter, and penalizing said counter by a predetermined penalty in response to a write to a block of data associated with said counter.

29. The method of claim 27 further comprising

identifying a least advantageous block of data for which a copy is stored in the cache memory, based on previously monitored accesses to blocks of data, and

as part of responding to a request for access to a desired storage device location for which a copy is not stored in the cache memory, retrieving from said direct access storage device a block of data including said desired storage device location, storing the block of data retrieved from said storage device, in place of the copy in said cache memory of said least advantageous block of data.

30. The method of claim 24 further comprising monitoring accesses to data for which copies are not stored in the cache memory.

45. A cache control circuit for a direct access storage device having a plurality of addressable locations, for controlling a cache memory having a total available memory capacity, storing copies of data retrieved from said direct access storage device, and a cache directory identifying data in said direct access storage device for which copies are stored in said cache memory, and identifying memory locations in said cache memory where each said copy is stored, the cache control circuit performing the steps of:

responding to a request for access to a storage device location for which a copy is stored in the cache memory, by accessing the copy stored in the cache memory,

responding to a request for access to a desired storage device location for which a copy is not stored in the cache memory, by accessing said desired storage device location from said storage device, and

responding to a change in said total available cache memory capacity by altering said cache directory.

46. The cache control circuit of claim 45 wherein in response to an increase in the total cache memory capacity, the cache control circuit modifies said cache directory to identify memory locations in said cache memory where copies of data from said direct access storage device may be stored, and then storing, in said cache memory, copies of data retrieved from said direct access storage device.

47. The cache control circuit of claim 45 wherein in response to a reduction in the total cache memory capacity, the cache control circuit modifies said cache directory to no longer identify memory locations that are not available in said cache memory.

48. The cache control circuit of claim 45 wherein said control circuit monitors accesses to data for which copies are stored in the cache memory.

49. The cache control circuit of claim 48 wherein said control circuit monitors accesses to data by maintaining a LRU queue in which data are ordered from most to least recently used.

50. The cache control circuit of claim 49 wherein said control circuit monitors accesses to data by maintaining statistics on types of accesses made to data.

51. (As Proposed to be Amended) The cache control circuit of claim 50 wherein said control circuit maintains statistics on types of accesses made to data by maintaining a counter associated with blocks of data, said counters being credited or penalized in response to types of accesses made to the associated block of data.

52. (As Proposed to be Amended) The cache control circuit of claim 51 wherein said control circuit maintains statistics by crediting a counter by a predetermined credit in response to a read to a block of data associated with said counter, and penalizing said counter by a predetermined penalty in response to a write to a block of data associated with said counter.

53. The cache control circuit of claim 51 wherein said control circuit

identifies a least advantageous block of data for which a copy is stored in the cache memory, based on previously monitored accesses to blocks of data, and

as part of responding to a request for access to a desired storage device location for which a copy is not stored in the cache memory, retrieves from said direct access storage device a block of data including said desired storage device location, and stores the block of data retrieved from said

storage device, in place of the copy in said cache memory of said least advantageous block of data.

54. The cache control circuit of claim 48 wherein said control circuit monitors accesses to data for which copies are not stored in the cache memory.

55. A program product, comprising:

(a) a program configured to perform a method of caching data for a direct access storage device having a plurality of addressable locations, comprising the steps of:

storing, in a cache memory having a total available memory capacity, copies of data retrieved from said direct access storage device, and a cache directory identifying data in said direct access storage device for which copies are stored in said cache memory, and identifying memory locations in said cache memory where each said copy is stored,

responding to a request for access to a storage device location for which a copy is stored in the cache memory, by accessing the copy stored in the cache memory,

responding to a request for access to a desired storage device location for which a copy is not stored in the cache memory, by accessing said desired storage device location from said storage device, and

responding to a change in said total available cache memory capacity by altering said cache directory, and

(b) a signal bearing media bearing the program.

56. The program product of claim 55, wherein the signal bearing media is a transmission type media.

57. The program product of claim 55 wherein the signal bearing media is a recordable media.